Automated military unit identification in battlefield simulation

Phillip D. Stroud and Ray C. Gordon
LAUR-97-849, SPIE proc. volume 3069, April 1997
Los Alamos National Laboratory, MS-F607, Los Alamos, NM 87545

ABSTRACT

It is the nature of complex systems, composed of many interacting elements, that unanticipated phenomena develop. Computer simulation, in which the elements of a complex system are implemented as interacting software objects (actors), is an effective tool to study collective and emergent phenomena in complex systems. A new cognitive architecture is described for constructing simulation actors that can, like the intelligent elements they represent, adapt to unanticipated conditions. This cognitive architecture generates trial behaviors, estimates their fitness using an internal representation of the system, and has an internal apparatus for evolving a population of trial behaviors to changing environmental conditions.

A specific simulation actor is developed to evaluate surveillance radar images of moving vehicles on battlefields. The vehicle cluster location, characterization and discrimination processes currently performed by intelligent human operators were implemented into a parameterized formation recognition process by using a newly developed family of 2D cluster filters. The mechanics of these cluster filters are described. Preliminary results are presented in which this GSM actor demonstrates the ability not only to recognize military formations under prescribed conditions, but to adapt its behavior to unanticipated conditions that develop in the complex simulated battlefield system.

1. INTRODUCTION

The modern theater-level battlefield is a complex, dynamic system that defies analytical approaches. It consists of thousands of interacting elements, including mounted and dismounted infantry troops, vehicles of many types, artillery, command posts of various levels, local and space-based sensor and communication systems, helicopters, aircraft, etc., as well as terrain, structures, and weather. Phenomena typically emerge in systems of this nature, which would never have been predicted, even with very elaborate models.[1,2] That complex systems exhibit emergent phenomena has two consequences of interest here. First, actor-based computer simulation is a natural tool for studying battlefield systems, providing a level of fidelity that can not be attained with analytic approaches. In actor-based simulation, simulation objects (representing entities on the real battlefield) are implemented on a distributed computer network, and they interact through asynchronous message passing. The second consequence is that cognitive abilities make up an essential aspect of many of the battlefield actors. Their ability to adapt their behavior to their perception of battlefield conditions is fundamental to what they do and how they interact with other elements. Static rule-based, connectionist, or algorithmic models are insufficient for building the software actors that represent elements with cognitive ability.

This paper describes the development of simulation actors that emulate a human operator's ability to recognize troop formations, including the ability to adapt to complex, dynamic battlefield conditions. The process of formation recognition has been automated through algorithms that evaluate and manipulate sets of 2D cluster filters. The behaviors invoked by an actor as it performs this formation recognition process is determined by the values of a set of control parameters. A new cognitive architecture has been developed by which an actor can adapt these control parameters to meet unanticipated external conditions. The two main components of the architecture are an evolutionary apparatus, which generates and maintains a population of trial cognitive behaviors, and a mechanism for evaluating the fitness of a trial cognitive behavior, based on an internal simulation of the external environment. A demonstration of real-time adaptation to dynamic battlefield conditions is presented.

2. BATTLEFIELD SIMULATION WITH THE JSTARS GSM

The Joint Surveillance Target Attack Radar System (JSTARS)[3] is an airborne sensor platform that flies in friendly airspace and uses Doppler radar to detect moving ground vehicles. Although it has a synthetic aperture radar (SAR) mode for obtaining higher resolution images of stationary structures, its primary utility is detecting moving tanks and armored personnel carriers. The system is able to cover large areas of a theater or battlefield, on the order of 100 km square. The radar return is down-linked to ground station modules (GSM). There is typically a GSM attached to each maneuver brigade. The brigade command post uses the JSTARS information to maintain situational awareness, and to direct fire support. These brigade level GSMs typically focus on control areas of 20-40 km in size. In addition, there is usually a GSM attached to division or corps headquarters. This GSM typically covers a larger area (scale size 40-80 km), and receives queuing messages from the brigade-level GSM's. The primary task of the GSM operator is to recognize and identify opposing force units moving within their assigned control areas. The GSM operator generally has a list of assigned tasks, based on current intelligence. A typical task might be to watch for a battalion strength unit moving towards a particular region of the battlefield (a "named area of interest") and then report its estimated time of arrival to the division headquarters. The GSM is one of many elements that comprise the complex battlefield system.

Battlefield simulation is developing into a valuable tool for training, mission planning, and assessing new systems and tactics.[4] The battlefield is an example of a complex system, which consists of many interacting elements. Collective phenomena emerge from these interactions, which produce an unpredictable, complex and dynamic environment for the elements within the system. In the simulation approach to studying such systems, the various elements that interact in a battlefield are simulated as software objects. These objects reside on a computer network, and interact with each other through asynchronous message passing (hence they are known as 'actors'). Battlefield simulations are composed of many types of actors, including tank and motorized rifle units (squads, platoons, companies, etc.), artillery units, command posts, opposing force units, the JSTARS platform, and JSTARS GSM's.

Each actor in a simulation has three segments: a physical state, a set of actions, and a cognitive segment. The physical state of an actor includes such data as its location and speed, its status, and the data it manipulates. For the GSM, the physical state includes the current moving target indicator (MTI) screen, a list of currently recognized units, and possibly the values of various parameters it uses to perform its tasks. The set of actions provides the methods by which an actor manipulates its own physical state and data, and the messages it can send, and handlers for messages it receive from other actors. The physical state and the collection of possible actions together form an object that is appropriate for object-oriented programming. The third segment, the cognitive segment, allows the actor to control its own actions, rather than having them invoked by external calls. The cognitive segment determines which actions are taken for various physical states, and can respond to changes in the physical state by appropriate actions. For the GSM actor, the cognitive segment must be able to find clusters on the MTI, recognize various military formations, and select appropriate actions to manipulate the list of recognized units and the MTI data, and decide on which messages to send.

3. A COGNITIVE ARCHITECTURE FOR ADAPTIVE ACTORS

Research in the fields of artificial intelligence has identified many of the essential cognitive elements, and various ways of combining them into cognitive architectures that allow machines to exhibit varying degrees of intelligence.[5,6] Some of these essential cognitive elements (memory, sensory components, perceptor, productions which invoke actions when the memory meets certain conditions, etc.) have been combined with the biological evolutionary view of intelligence[7] to develop a cognitive architecture that extends the capability of the cognitive segment of simulation actors. The methodology originated in an off-line process for automated behavior discovery in flight and fire controllers of an airborne laser simulation actor.[8] That off-line methodology has now been internalized into the actor to give real-time or on-line adaptive capability.

A schematic of this architecture is shown in Fig. 1. The actor's cognitive segment is divided into two main parts: an autonomous behavior which performs low-level cognition; and a higher level mechanism that can adapt the low-level behavior to changing environmental conditions. In non-adaptive actors, the cognitive segment consists only of the autonomous behavior, which, along with the actor's physical state and collection of actions, forms the actor . The autonomous behavior relating actions to physical state can be implemented as a collection of algorithms, a control system, a simple set of rules, an elaborate set of rules known as an expert system, or a neural network of various configurations.

Whatever the implementation of the autonomous behavior, there will be a set of adjustable control parameters that determine the behavior (e.g. the set of connection weights for a neural network representation). In non-adaptive actors, the adjustable control parameters that control the actor's autonomous behavior are pre-programmed (or pre-trained) to build in expertise. The high-level component of this cognitive architecture provides a mechanism for the actor to adapt its low-level autonomous behavior by adjusting the control parameters. In the GSM actor, it is useful to be able to extract the current strategy, so the autonomous behavior set is represented as a parameterized rule set, where the parameters that determine the behavior have easily extractable meaning.

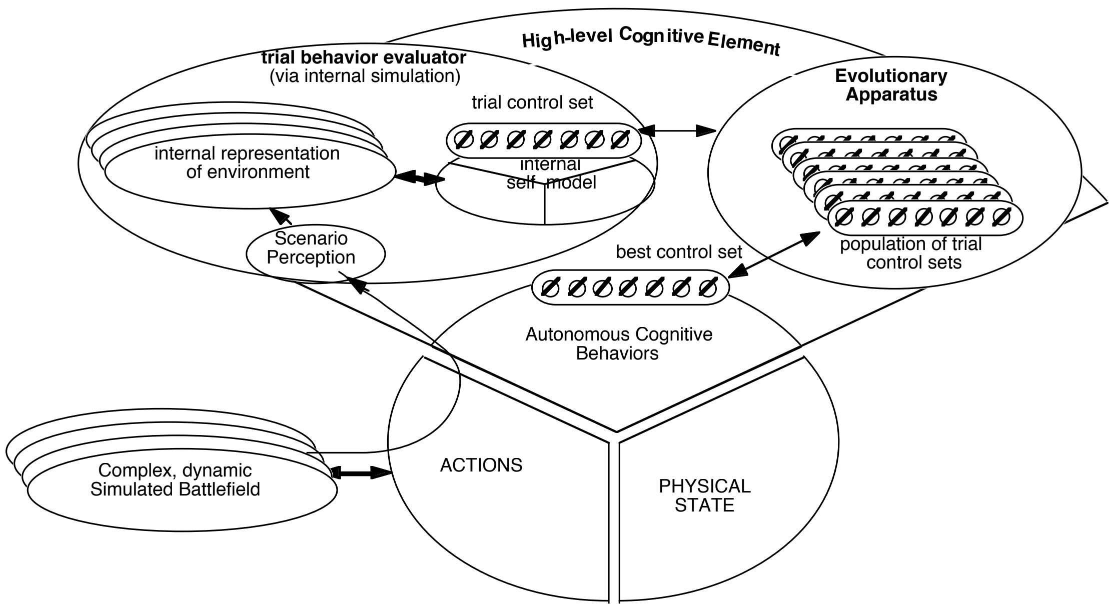

Fig. 1. The cognitive architecture that extends the pre-programmed autonomous cognitive element to higher-level cognition.

Two primary elements make up the high-level cognitive part: a mechanism that can estimate the effectiveness of trial behaviors using an internal representation of the external world, and an evolutionary apparatus. The internal representation adjusts itself to correspond to currently perceived external conditions. For the GSM actor, the internal representation has an element that generates (and stores in memory) random MTI screens that are consistent with the currently perceived battlefield conditions (expected number of battalions, current visibility conditions, terrain, weather, expected opposing forces tactics, etc.), which are stored as part of the actor's physical state data. The perceived external conditions can be changed by direct sensors, or by messages passed from the command post to which the GSM is attached. There is also an internal representation of the actor's low-level autonomous behavior, and a means of evaluating how well a trial autonomous behavior (represented by a trial set of control parameters) would perform against the currently perceived external conditions. Since the fitness of a given behavior set will have a large stochastic component, the internal fitness estimation must evaluate a trial behavior over many possible cases to obtain a valid fitness estimate. In the case of the GSM, the fitness estimation requires hundreds of independent random MTI screens.

The evolutionary apparatus generates and maintains a population of alternative behavior sets, in a working memory dedicated to the higher level cognition. The collection of all possible behaviors (represented by all possible control parameter settings) evaluated in all possible environments consistent with perceived conditions, provides a well-posed problem for the genetic algorithm.[9] The autonomous behavior produced by the set of control parameters is encoded into a chromosome. A population of chromosomes is evolved by the evolutionary apparatus, using genetic cross-over, mutation, and fitness based selection. The evolutionary apparatus uses the internal representation to evaluate various trial behaviors. The evolutionary process occurs in a thread in the actor. When a new behavior set is found that performs better than the current autonomous behavior set in internal evaluation against the internal representation, that new, better adapted behavior is then used for the autonomous behavior.

4. THE AUTONOMOUS COGNITIVE ELEMENT OF THE GSM ACTOR

The starting point for developing an actor that can emulate a human operator is the training that the human operators receive. GSM operators currently undergo a U.S. Army training program, which provides a model for the formation recognition process.[2] This process is seen to consist of three steps. The first step is to locate clusters of blips on the MTI screen. The 2D moving target indicator screen contains on the order of a million pixels, of which on the order of a hundred may be lit. While cluster location is easily accomplished by the human eye and brain, the complexity of this process becomes apparent when automation is attempted. The second step is to select and characterize the cluster of blips. The GSM operator selects a cluster manually with a mouse selection box. The choice of exactly where to put the selection box is a cognitive decision-making process. There are then several on-screen buttons which cause the MTI to characterize the selected cluster, as to the number of vehicles or blips, their speed and heading, whether they are tracked or wheeled vehicles. The actual characterization of a cluster is a mechanical rather than cognitive process. The third step is to decide whether the cluster is a military unit of interest. This is accomplished by applying a set of rules to the cluster characteristics. These rules can be cast into standard productions of the form IF (condition c) THEN (perform action A). The conditions are based on cluster characteristics such as the number of wheeled blips, the size of the cluster, etc. An example production in a search for battalions might be: IF (total number of blips in cluster is less than 16) THEN (mark this cluster as "not a battalion", and look for another cluster).

The autonomous behavior of the GSM actor takes a list of blip locations and generates a list of formations which it recognizes. It manipulates data, performs searches, and makes decisions, but it does not learn. Any adaptation it performs has to be pre-programmed (as in IF condition x is perceived, then use strategy s_x, otherwise use strategy s_0). It has a set of behaviors which are invoked according to the current physical state of the GSM. It is constructed on the three steps which the human operator uses to recognize formations. There is a process for searching for clusters of blips, based on a set of translatable 2D filter functions. There is a process for characterizing a cluster once it is located, and finally there is a process for deciding whether a cluster is a battalion or other formation. When a battalion is found, it is added to a list of recognized battalions, and the corresponding blips are removed from the list of blips. The entire process of recognizing battalions is subject to a set of control parameters, such as the number of steps to take in searching for a cluster before evaluating the cluster.

Opposing forces units may be deployed in a wide variety of formations. The lowest level formation is the platoon, which can consist of three or four squads (one vehicle, tank or armored personnel carrier per squad), in line, column, two-up-one-back, chevron-left or right, echelon-left or right, bounding-over-watch, or other formation. There is a similar variety of possible formations for platoons within companies, and companies within battalions. While each of these formations has a doctrinal position for each sub-unit, the actual positions of the vehicles on the battlefield will deviate from the doctrinal position. These deviations can depend on many factors which change with the dynamic battlefield. In addition, there are atmospheric phenomena, line-of-sight blockage, etc. that prevent some vehicles from being detected by the JSTARS. There are thus hundreds of possible doctrinal formations for a battalion, and a possibly wide range of variation within each one. In addition, a formation can be moving in any direction. The formation recognition process must be able to handle the arbitrary orientation of formations.

4.1 Complete set of ortho-normal 2D cluster filters.

A family of sets of filter functions has been developed to search for and characterize clusters in 2D data. These filters are formed by the product of a 2D Gaussian and a set of 2D polynomials. The scale size and location of the Gaussian can vary, and the polynomials are defined in terms of the scale size and location of the Gaussian. There is a complete set of orthogonal 2D filters for every triplet of values of three adjustable parameters: x_0 and y_0 (giving the location of the center of the Gaussian in the 2D plane), and d (the standard deviation of the Gaussian). The terms in the polynomials contain powers of the transformed variables $\xi=(x-x_0)/d$ and $\eta=(y-y_0)/d$. The shape of a filter function is invariant, but its location and scale size changes according to the values of x_0, y_0 and d.

The 2D Gaussian (normalized by a $\pi^{1/2}$ factor to simplify the polynomials) is given by $g(\xi, \eta)= \exp[-(\xi^2+ \eta^2)/2]/\pi^{1/2}$. If the j^{th} polynomial is given by $V_j(\xi, \eta)$, then the j^{th} cluster filter will be $S_j(\xi, \eta)=V_j(\xi, \eta)g(\xi, \eta)$. If the set of filter functions is orthogonal, then single filter coefficients and combinations of a few coefficients can have direct physical interpretations as cluster characteristics. The filter functions are normalized to unity. The ortho-normal condition on the set of cluster filters indicates that the filters can be found by obtaining a set of 2D polynomials that are ortho-normal with respect to a weight function of $\exp[-(\xi^2+\eta^2)]/\pi$.

$$\int_{-\infty}^{\infty} d\xi \int_{-\infty}^{\infty} d\eta S_k^*(\xi,\eta) S_{k'}(\xi,\eta) = \int_{-\infty}^{\infty} d\xi \int_{-\infty}^{\infty} d\eta V_k^*(\xi,\eta) V_{k'}(\xi,\eta) e^{-(\xi^2+\eta^2)}/\pi = \delta_{kk'}. \tag{1}$$

A 1D set of orthogonal functions based on 1D Gaussian weighting has been used extensively in time-domain signal processing applications.[10-15] In the 1D case, the filter functions are composed of a 1D Gaussian multiplied by Hermite polynomials. Several applications have been published using filters consisting of a 2D Gaussian multiplied by 2D polynomials formed from the cross product of a Hermite polynomial in ξ and another Hermite polynomial in η.[16-18] In this crossed-Hermite approach the are, for example, three second order filters, which have polynomials given by $H_0(\xi)H_2(\eta)$, $H_1(\xi)H_1(\eta)$, and $H_2(\xi)H_0(\eta)$, where the Hermite polynomials are $H_n(\xi)=\{1, 2\xi, 4\xi^2-2, ...\}$. This does produce a complete set of orthogonal filter functions, which has been applied to the physiology of the eye and to machine vision.

An alternative orthogonal set is developed here that has rotational invariance properties not possessed by these crossed-Hermite filters. There are characteristics of clusters that are independent of the orientation of the cluster, such as ellipticity of a bounding ellipse, "triangularity", or central tendency. Rotational symmetries are therefore a useful criteria for selecting a particular set of orthogonal filter functions out of the infinite number of possible sets. The 2D polynomials are first separated into radial and angular functions.

$$V_j(\xi,\eta) = V_j(\rho\cos\theta, \rho\sin\theta) = R_{n,m}(\rho)\Theta_m(\theta) \tag{2}$$

where $\rho = \sqrt{\xi^2+\eta^2}, \xi = \rho\cos(\theta), \eta = \rho\sin(\theta)$. A slightly modified version of Born and Wolf's derivation of the Zernike polynomials[19] will allow the determination of the radial and angular functions. The orthogonal set containing angular functions with every m-fold rotational symmetry is given by the set for all integer m ≥ 0 of

$$\Theta_m(\theta) = \begin{matrix} 1 & \text{for } m=0 \\ \sqrt{2}\cos(m\theta) & \rangle \text{for } m>0 \\ \sqrt{2}\sin(m\theta) & \end{matrix} \tag{3}$$

In this formulation of Θ, it has been assumed that only real valued functions are of interest here. To allow for expansion of complex functions onto these filter functions, angular functions of the form $\exp(\pm im\theta)$ would be used instead. The rest of the following would be unchanged, except for the normalization and indexing scheme. For a given value of m, $R_{n,m}(\rho)$ is then a polynomial in ρ of order n, containing only terms with powers m, m+2, ..., n. m and n are either both even or both odd. For a given n, the allowed values of m are n, n-2, n-4,..., down to 0 or 1. For a given value of n, if m is 0, there is one filter function, and when m is not 0 there are two filter functions, one using the sine and one using the cosine. There are thus a total of n+1 filter functions with a given value of n (i.e. of order n). This restriction on powers of terms contained in $R_{n,m}$ allows the conversion $R_{n,m}(\rho)=t^{m/2}Q_k(t)$, where $t=\rho^2$. $Q_k(t)$ is then a 1D polynomial of order k which contains terms of all integer powers from 0 through k. k is related to n and m by $k=(n-m)/2$. The orthogonality condition (1), along with the definition of the angular functions, gives a requirement for Q(t):

$$\int_0^{\infty} dt\, t^m e^{-t} Q_k(t) Q_{k'}(t) = \delta_{kk'} \tag{4}$$

The set of orthogonal polynomials that satisfies (4) is the set of generalized Laguerre polynomials[20] , appropriately normalized to $Q_k(t) = \sqrt{\frac{((n-m)/2)!}{((n+m)/2)!}} L_k^m(t)$. The coefficient of the i^{th} power of t in the generalized Laguerre polynomial $L_k^m(t)$ is given by $(-1)^i(k+m)!/[(m+i)!(k-i)!i!]$. The radial polynomials, in terms of these generalized Laguerre polynomials, are simply

$$R_{n,m}(\rho) = \sqrt{\tfrac{((n-m)/2)!}{((n+m)/2)!}}\rho^m \; L^m_{(n-m)/2}(\rho^2) \tag{5}$$

The complete set of ortho-normal Gaussian weighted 2D polynomial filters with orientation invariance properties is

$$S_j(\rho\cos\theta, \rho\sin\theta) = \sqrt{\tfrac{((n-m)/2)!}{\pi((n+m)/2)!}}\rho^m \; L^m_{(n-m)/2}(\rho^2)\,\Theta_m(\theta)\exp(-\rho^2/2) \tag{6}$$

A similar result, cast as generalized Laguerre polynomials with argument $\xi\eta$ instead of ρ^2, was given by Dodonov and Man'ko.[22] Their orthogonal set has symmetries with respect to reflection through the $\xi=\eta$ line, rather than the rotational symmetries desired here.

The Zernike indexing convention[21] relating j, m and n can be adopted. For m=0, the index is j=n(n+1)/2+1. For m≠0, there are two functions for each n,m pair, one with index j=n(n+1)/2+m and the other with j=n(n+1)/2+1+m. The one with the odd j index takes the sin(mθ) angular function, and the one with the even j takes the cos(mθ) angular function. The first eleven polynomials are given in Table 1, in both polynomial form $V_j(\xi,\eta)$, and as separated radial and angular functions, $R_{n,m}(\rho)\Theta_m(\theta)$. The set of cluster filters is obtained by multiplying the polynomials by the Gaussian, $S_j(\xi,\eta)=V_j(\xi,\eta)g(\rho)$.

j	n	m	$V_j(\xi,\eta)$	$=R_{n,m}(\rho)\Theta_m(\theta)$
1	0	0	1	$=1$
2	1	1	$\sqrt{2}\xi$	$=\sqrt{2}\rho\cos(\theta)$
3	1	1	$\sqrt{2}\eta$	$=\sqrt{2}\rho\sin(\theta)$
4	2	0	$1-\xi^2-\eta^2$	$=1-\rho^2$
5	2	1	$2\xi\eta$	$=\rho^2\sin(2\theta)$
6	2	1	$\xi^2-\eta^2$	$=\rho^2\cos(2\theta)$
7	3	1	$2\eta-\xi^2\eta-\eta^3$	$=(2\rho-\rho^3)\sin(\theta)$
8	3	1	$2\xi-\eta^2\xi-\xi^3$	$=(2\rho-\rho^3)\cos(\theta)$
9	3	3	$\sqrt{3}\eta\xi^2-\eta^3/\sqrt{3}$	$=(\rho^3/\sqrt{3})\sin(3\theta)$
10	3	3	$\xi^3/\sqrt{3}-\sqrt{3}\xi\eta^2$	$=(\rho^3/\sqrt{3})\cos(3\theta)$
11	4	0	$1-2\xi^2-2\eta^2+\xi^4/2+\eta^4/2+\xi^2\eta^2$	$=1-2\rho^2+\rho^4/2$

Table 1. The first 11 orthogonal cluster polynomials.

This set of cluster functions forms a complete orthogonal set over the 2D plane. There is a one to one correspondence with the Zernike functions, where the corresponding functions have the same angular dependence. (The 2D Zernike polynomials form a complete orthogonal set over a unit circle, with a constant weighting function, rather than over the 2D plane with a Gaussian weighting.) An optical interpretation can accordingly be given to the various filter functions. The S_2 and S_3 filters can be identified with x and y tilts or offsets, the S_4 filter can be identified with defocus or spread, the S_5 and S_6 filters can be identified with astigmatism or ellipticity, and the S_7 and S_8 filters can be identified with coma.

Any function in the x-y plane can then be expanded onto a complete set of ortho-normal cluster filters:

$$f(x,y) = \frac{1}{d}\sum_{j=1}^{\infty} c_j S_j\left(\frac{x_i - x_0}{d}, \frac{y_i - y_0}{d}\right) \tag{7}$$

where the expansion coefficients are

$$c_j(x_0,y_0,d) = \frac{1}{d}\iint dx\,dy\,f(x,y)\,S_j(\frac{x_i - x_0}{d},\frac{y_i - y_0}{d}) \tag{8}$$

If the MTI pixels are either on or off (indicating the detection or non-detection of a vehicle at the corresponding location), the MTI intensity distribution can be represented as a sum of Dirac functions, one for each of N blips, where the ith blip is at location (x_i, y_i) for i=1,...,N: $f(x,y) = \sum_{i=1}^{N} \delta(x - x_i, y - y_i)$. In this case, the expansion coefficients are obtained by a sum over the N blips:

$$c_j(x_0,y_0,d) = \frac{1}{d}\sum_{i=1}^{N} S_j(\frac{x_i - x_0}{d},\frac{y_i - y_0}{d}) \tag{9}$$

It is much more efficient to sum over actual blips with (9), than to perform the weighted sum over all pixels required by the formulation of (8). Each triplet {x_0, y_0, d} has a corresponding complete set of ortho-normal cluster filters.

4.2 Search for clusters of size d.

Suppose the data f(x,y) contains a cluster of some size D (e.g. the rms distance from the cluster centroid to the blips in the cluster), containing n blips, with centroid located at (x_c, y_c). The set of cluster filters with x_0 and y_0 set to the cluster centroid, and d~D, will then have c_2 and c_3 nearly equal to 0. The contribution of blips outside the cluster to these coefficients is very small because of the exponential factor in the filters. For the blips in the cluster, the contribution to c_2 of those blips to right of the centroid ($x_i>x_0$) tend to cancel that of the blips to the left ($x_i<x_0$). Similar cancellation of contribution occurs in the y direction for c_3. If the cluster filter location is some distance away from the cluster centroid, but far from other data points not belonging to the cluster, it is easy to see that the cluster centroid can be estimated by

$$\begin{aligned} x_c &= x_0 + c_2(x_0,y_0,d)/\sqrt{2}c_1(x_0,y_0,d) \\ y_c &= y_0 + c_3(x_0,y_0,d)/\sqrt{2}c_1(x_0,y_0,d) \end{aligned} \tag{10}$$

This suggests a simple iterative cluster search algorithm in which 1) a filter set is created with some scale size, d, and random center location (x_0, y_0); 2) the coefficients c_1, c_2, and c_3 are evaluated for this filter set with the data; 3) a cluster centroid location is estimated using (10); 4) the set of filters is moved to $x_0=x_c$ and $y_0=y_c$; and 5) go back to step 2. Several loop exit criteria can be used: loop until c_2 and c_3 are sufficiently close to 0, or a preset number of loops have been tried, or a preset number of floating point operations have been expended. Even one blip prevents c_1 from taking a value of 0, although it can become very small, and must be checked for under-flow before dividing.

This method has been found by computer simulation to be extremely efficient and effective at locating clusters. The search is tuned to find clusters of any desired size range by selecting the value of d. The method will locate the center of a cluster of size ~d even it is formed from a group of smaller sub-clusters . On the other hand, the method will zero in on a cluster of size ~d even if this cluster is one of many clusters in a larger super-cluster. The method fails when clusters of size ~d are separated by distances also on the order of d. In this case, the search can get stuck in between clusters. This situation can be avoided by searching for and characterizing larger clusters first.

4.3 Characterization of a cluster

Once a centroid has been found, (i.e. the filter location parameter is such that the tilt coefficients c_2 and c_3 are zero), the size, shape and number in the cluster can be evaluated through the second order filter coefficients, c_4, c_5, and c_6. These three coefficients can be used to determine an ellipse that best characterizes the cluster. Suppose a cluster of n blips could be described by an elliptical bi-Gaussian distribution, with semi-major standard deviation a, semi-minor standard deviation b, and orientation angle θ(specifying the angle from the x-axis counter-clockwise to the major axis of the ellipse. This cluster distribution is given by

$$G(x,y;n,a,b,\theta)=\frac{n}{2\pi ab}\exp\left\{-\frac{\left[(x-x_0)\cos\theta-(y-y_0)\sin\theta\right]^2}{2a^2}-\frac{\left[(y-y_0)\cos\theta+(x-x_0)\sin\theta\right]^2}{2b^2}\right\} \quad (11)$$

The constant-density contours of this distribution are ellipses, with the ratio of major to minor axis given by a/b, and the orientation of the major axis specified by its angle counter-clockwise from the x axis, θ. The ellipse centered at x_0, y_0, with semi-major axis a, and semi-minor axis b, with orientation angle θ, contains $1\text{-}e^{-1/2} = 39.35\%$ of the distribution.

The procedure described in Section 4.2 is used to move x_0 and y_0 to the center of the cluster. Then, the second-order filters are evaluated. For an ideal elliptical Gaussian cluster, the coefficients of the second-order cluster filters can be found analytically:

$$\begin{aligned} c_1 &= n\Big/\sqrt{\pi\left(1+a^2/d^2\right)\left(1+b^2/d^2\right)} & c_4 &= \frac{c_1\left(1-a^2b^2/d^4\right)}{\left(1+a^2/d^2\right)\left(1+b^2/d^2\right)} \\ c_5 &= \frac{c_1\sin(2\theta)\left(a^2/d^2-b^2/d^4\right)}{\left(1+a^2/d^2\right)\left(1+b^2/d^2\right)} & c_6 &= \frac{c_1\cos(2\theta)\left(a^2/d^2-b^2/d^4\right)}{\left(1+a^2/d^2\right)\left(1+b^2/d^2\right)} \end{aligned} \quad (12)$$

The parameters describing the elliptical Gaussian distribution can be obtained from the second-order filter coefficients by inverting (12), giving

$$a=d\sqrt{\frac{c_1-c_4+\sqrt{c_5^2+c_6^2}}{c_1+c_4-\sqrt{c_5^2+c_6^2}}}\qquad b=d\sqrt{\frac{c_1-c_4-\sqrt{c_5^2+c_6^2}}{c_1+c_4+\sqrt{c_5^2+c_6^2}}}\qquad \theta=\arctan2(c_5,c_6)/2 \quad (13)$$

Rather than evaluate the arctan for θ, the values of $\sin(\theta)$ and $\cos(\theta)$ (which together characterize the orientation of the ellipse) can be calculated from the coefficients without any trig evaluations, for a slight gain in computational efficiency. The coefficients c1 and c4 can be used by themselves to estimate the area of the cluster under the approximation that the ellipticity was small. This area estimate, $\pi ab = \pi d^2(c_1-c_4)/(c_1+c_4)$, is independent of the orientation of the cluster.

An actual cluster of blips may not be well described by an elliptical Gaussian distribution, but the cluster length, width, and orientation can be characterized by the a, b, and theta values derived from the coefficients produced by the cluster. A crisp ellipse can be assigned to the cluster, by multiplying a and b by a crisping factor. A blip inside this crisp ellipse will be counted as part of the cluster, while blips outside the cluster will not. This method produces an ellipse for a given cluster that is independent of the value of the filter scale parameter d over a surprisingly large range of d. An appropriate value for the crisping factor will depend on the nature of the clusters. If clusters tend to have outliers attached to them, a larger crisping factor is appropriate. On the other hand, if clusters have sharp boundaries, or occur in close proximity to other clusters, a smaller crisping factor is appropriate. The crisping factor is a parameter that can be adapted to battlefield conditions.

Higher order filters can be used to further characterize the clusters. The four third-order filter coefficients (c_7 through c_{10}), for example, can be used to characterize the “triangularity” of a cluster. This would allow the GSM to distinguish between a battalion with three companies in a two-up-one-back formation, and a battalion with a four companies in a square formation, based on four filter coefficients. The evaluation of higher-order filters requires relatively few additional computations. The evaluation of one filter requires one exponentiation and a few multiply and adds for each of the N blips in the list of blips. Evaluation of an additional filter (different j, same x_0, y_0, and d) does not require the exponentiation to be repeated, but only a few additional multiply and adds per blip to compute the additional polynomial value. The ξ–η polynomial form can be used to avoid the relatively expensive trig function evaluations. The computational requirement for locating and characterizing a cluster is linear in the number of blips.

4.4 Evaluation of the cluster

Once a cluster has been located, and had its length, width, orientation, and number of blips characterized, it must be evaluated to determine whether it was produced by a military formation, and if so, by what military formation. This recognition process is implemented as a parameterized rule set. The rules contain adjustable parameters. The appropriate value to take for these parameters depends on many factors, such as visibility, terrain roughness, whether units are at full strength, or have been attrited or reinforced, and the tactics in current use. A simple rule set that mimics the expertise provided by the basic GSM operator training for recognizing battalions is:

IF (width of cluster > 2.5 km) THEN reject cluster as battalion
IF (length of cluster > 4.0 km) THEN reject cluster as battalion
IF (number in cluster > 80) THEN reject cluster as battalion
IF (number in cluster < 25) THEN reject cluster as battalion
IF (more than 10% of blips in cluster differ from cluster speed by more that 10 %) THEN reject cluster as battalion
IF (more than 10% of blips in cluster differ from cluster heading by more that 20°) THEN reject cluster as battalion
IF (cluster is split by a terrain feature) THEN reject cluster as battalion
IF (all other rules fail to reject cluster) THEN accept cluster as a battalion

4.5 The parameterized autonomous formation recognition process

The pieces can now be assembled into an algorithm for locating, characterizing and discriminating all clusters in a 2D MTI screen. The first step is to get the moving target indicator image. The second step is to convert the 2D pixel array into a list of blips, each with an x and y location, and put the blip list into working memory. This can include pre-filtering to remove isolated blips. The third step is to perform a series of cluster evaluations using the cluster filters. The last step is to report any recognized formations. The third step is accomplished by looping over six sub-processes consisting of 1) initializing the filter to a random location and scale size within preset limits, 2) finding a cluster centroid by iterative refinement of the filter location based on c_2 and c_3, 3) characterizing the cluster length, width, orientation and number using c_4, c_5, and c_6, 4) deciding whether the cluster is a battalion by evaluating the rule-based production system, and updating the list of recognized battalions if necessary, 5) removing the cluster from the blip list, if appropriate conditions are met, and 6) determining if the search is finished.

The autonomous formation recognition algorithm has several adjustable controlling parameters. The number of search steps allowed to locate the cluster centroid can take values from 1 and up. The initial filter location is taken as a uniformly distributed random variate within the MTI control area, but alternate distributions of starting locations could be used. The filter scale parameter is initialized prior to each cluster search from a uniform random variate on the interval d_{min} to d_{max}, so that d_{min} and d_{max} are adjustable control parameters. The performance of the formation recognition process depends sensitively on the crisping factor γ. Appropriate values range from 1 to 3, depending on the nature of the clustered data. The minimum and maximum number of blips in a cluster that will be called a battalion can also be considered control parameters, in that these two numbers have a strong effect on the formation recognition process. There are also some constraint parameters. For example, the total number of floating point operations which are allocated to the GSM to perform an evaluation of the MTI might be fixed by external factors such as the required turn-around time.

The behavior of the autonomous cognitive element algorithm, i.e. the process by which it evaluates the MTI, is determined by the values of these control parameters. For a proof-of-principle, a set of seven parameters were selected: (the minimum filter scale size, the maximum filter scale size, the maximum battalion area, the ellipse crisping factor, the minimum number of blips to qualify as a battalion, the maximum number of blips to qualify as a battalion, the maximum number of search steps before characterizing the cluster and restarting the filter). There can obviously be many more parameters.

5. INTERNAL FITNESS EVALUATION

As described in section 3, the GSM actor has a means of generating trial behaviors. Since the behavior is specified by the seven parameter values, creation of a trial behavior is simply a matter of generating trial values for the seven control parameters. The GSM actor also has, within its cognitive structure, a mechanism for estimating the fitness or performance of a given behavior under the currently perceived external conditions. This mechanism works as follows.

First, a random set of military formations consistent with the currently perceived environment is generated and placed within the actor's working memory. These units are then "pinged" using an internal JSTARS model, accounting for the currently perceived weather, terrain, and visibility conditions, to produce an internal blip list, which is also put in the GSM actor's internal working memory. With only a slight semantic stretch, it can be said that the GSM actor imagines some military units on an imagined battlefield, figures out what the resulting MTI could look like, and holds the resulting blip list in its mind. The other actors in the external simulated battlefield do not know about any of this: it is all internal to the GSM actor. Also in its mind, it has a model of its own behavior, as specified by the seven trial control parameters. This internal model of its own formation recognition process is then applied to the imagined blip list, and the result is a list of battalions, which is also stored as an object in the working memory of the GSM actor. The list of recognized battalions can then be compared with the list of military formations that were first imagined. This comparison gives a measure of fitness for the trail set of run-time GSM parameters. To obtain a statistically meaningful estimate of this fitness, this whole process is repeated a number of times with independent internal (imaginary) sets of military units, and the average performance over this ensemble is taken as the fitness estimate of the trial set of control parameters.

The performance of a run-time GSM against a particular set of vehicles is found by comparing the list of recognized battalions against the list of military units used to generate the blip list, as follows. Each battalion in the recognized battalion list is paired up with each battalion in the actual battalion list, and their separation is calculated. The recognized-actual battalion pair with the smallest separation is then scored by how large the separation is. For zero separation, this pair contributes no recognition errors. For very large separation, the pair would contribute two recognition errors (one actual battalion missed plus one false alarm). A simple scoring formula adds $2/[1+(s_w/s)^2]$ recognition errors when the recognized battalion is a distance s away from the actual battalion. The parameter s_w is the error distance at which one recognition error would be counted, whether or not the battalion were added to the recognized battalion list. A value of 6km was selected for s_w. After the closest pair is scored, the actual and recognized battalions are removed from their respective lists. The process is then repeated for the remaining battalions, accumulating the recognition errors. When one list is empty, one recognition error is added for each remaining unit in the other list. The accumulated number of recognition errors is the measure of performance of a behavior for a particular MTI screen.

6. STRATEGY ADAPTATION BY GENETIC ALGORITHM

A genetic algorithm is used to search for better-adapted autonomous behaviors. Seven genes are used to represent the seven control parameters. Each gene is represented as a 7 bit binary string, allowing 128 possible values for each. The seven genes are concatenated into a 49 bit chromosome. The 2^{49} possible chromosomes then represent $5.6(10)^{14}$ alternative formation recognition strategies. A gradient decent search strategy does not work because of the existence of many local optima, and lack of a derivative of the fitness with respect to the parameter or chromosome values. The genetic algorithm approach works as follows.

First, the internal representation used to evaluate trial control parameters is tied into the currently perceived external conditions. An initial population of trial chromosomes is constructed by copying the original (current best) chromosome, and mutating a few bits at random. A population size of 24 chromosomes is appropriate for a 49 bit chromosome. The next step is to estimate the fitness of each of these trial chromosomes, by running internal models of their resulting run-time GSM behaviors against hundreds of internally generated MTI screens. The next step is to evolve the population by stepping through generations, looping over the four processes of 1) select parents from the population, based on their estimated fitness 2)breed new trial chromosomes, using genetic cross-over and mutation, 3) estimate the fitness of the trial chromosomes, and 4) sort new chromosomes into the population, replacing less fit chromosomes. Whenever a chromosome is found that has better estimated fitness than the current autonomous behavior, the control parameters are replaced with the better values.

7. DEMONSTRATION

A C++ object-oriented implementation of a JSTARS GSM simulation actor has been developed for use in the Los Alamos battlefield simulation environments SAMSON, JOINTSIM, and JWARS. The autonomous behavior is initialized to a baseline behavior that gives a first approximation to the training received by GSM operators. The baseline set of control parameters is (minimum filter scale size = 0.3 km, maximum filter scale size = 0.5 km, maximum battalion area = 20 square km, ellipse crisping factor = 2.0, minimum number of blips to qualify as a battalion = 18, maximum number of blips to qualify as a battalion = 38, maximum number of steps per cluster search = 45). In the initial battlefield environment, one battalion is expected in the MTI coverage area, the formation is exactly given by doctrine, and the visibility is perfect. The autonomous behavior is limited to a maximum of 20,000 trig function evaluations per MTI screen evaluation. The fitness estimator is set to use 200 random MTI screens to estimate the fitness of trial behaviors. The baseline behavior is evaluated on the 200 MTI screens, and the fitness is found to be an average of 0.0299 recognition errors per screen. The evolutionary apparatus then generates and evolves a population of 24 trial behaviors. In less than two minutes, on an 80Mhz Power Macintosh, 144 trial behaviors were generated and evaluated, and a new behavior was found which gives a fitness of 0.00527 recognition errors per screen (more than five times better). The new control parameters are (minimum filter scale size = 0.85 km, maximum filter scale size = 2.43 km, maximum battalion area = 15.2 square km, ellipse crisping factor = 1.77, minimum number of blips to qualify as a battalion = 21, maximum number of blips to qualify as a battalion = 62, maximum number of steps per cluster search = 99). These parameters give an autonomous GSM behavior that is effective for one expected battalion in a 40x25 km area, when the formations are not significantly different from doctrinal.

At some later time, however, the battlefield conditions have changed so that there are expected to be four battalions, two independent companies, and 10 independent squads in the battlefield. In addition, the visibility drops to 70% due to battlefield haze, and the formations are disrupted to the degree that within a unit, every sub-unit is off of its doctrinal position by up to 40% of the size of the unit (battalions within brigades or regiments, companies within battalions, platoons within companies, and squads within platoons). The previous behavior is evaluated on 200 MTI screens consistent with the new conditions, and the fitness is found to be an average of 0.743 recognition errors per screen. The evolutionary apparatus then evolves its population of 24 trial behaviors. In 25 minutes, on an 80Mhz Power Macintosh, 100 trial behaviors are evaluated, and a new behavior found which gives a fitness of 0.188 recognition errors per screen. The new control parameters are (minimum filter scale size = 0.42 km, maximum filter scale size = 1.11 km, maximum battalion area = 15.0 square km, ellipse crisping factor = 2.81, minimum number of blips to qualify as a battalion = 18, maximum number of blips to qualify as a battalion =56, maximum number of steps per cluster search = 107). These parameters give an autonomous GSM behavior that is better adapted to the new battlefield conditions.

At some later time yet, the battlefield conditions again change so that there are expected to be three battalions moving as part of a regiment plus another independent battalion. The visibility has dropped to 50% due to battlefield haze, and the formations are disrupted to the degree that within a unit, every sub-unit is off of its doctrinal position by up to 20% of the size of the unit. The previous behavior is evaluated on 200 MTI screens consistent with the new conditions, and the fitness is found to be an average of 2.256 recognition errors per screen. The evolutionary apparatus then evolves its population of 24 trial behaviors. In 34 minutes, on an 80Mhz Power Macintosh, 135 trial behaviors are evaluated, and a new behavior found which gives a fitness of 0.9825 recognition errors per screen. The new control parameters are (minimum filter scale size = 0.14 km, maximum filter scale size = 0.30 km, maximum battalion area = 10.2 square km, ellipse crisping factor = 1.61, minimum number of blips to qualify as a battalion = 8, maximum number of blips to qualify as a battalion =46, maximum number of steps per cluster search = 111). Even though the autonomous GSM behavior has adapted to the new battlefield conditions, it does not take the obvious strategy of locating regiments before battalions. This points out the necessity to be able to represent a vast variety of alternative behaviors .

The old and evolved control parameter sets can be tested against other set of internal MTI screens. Independent sets of 200 screens were generated, and used to determine the statistical variance in the estimate of the fitness of behaviors. For 200 screens, the error in the estimate is negligible compared to the differences in fitness of different behaviors. The behaviors are adapting to the new environment, not merely learning a training set of MTI screens. The use of internal test sets to reevaluate trial chromosomes can provide a mechanism to adjust the number of MTI screens needed to provide a reliable fitness estimate.

8. CONCLUSION

It is the nature of complex systems, composed of many interacting elements, that unanticipated phenomena develop. Computer simulation, in which the elements of a complex system are implemented as interacting software objects, is an effective tool to study these complex systems. A new cognitive architecture has been developed for constructing simulation actors that can adapt to external conditions that were never considered by the actor builders. The cognitive architecture extends non-adaptive and traditional adaptive architectures (in which a set of external parameters is sensed, and the behavior of the actor is adjusted according to these parameters in a pre-programmed way). The new cognitive architecture generates trial behaviors in its mind, estimates their fitness using an internal representation of the system, and uses an evolutionary apparatus to continuously adapt a set of trial behaviors to environmental conditions.

This architecture has two application-specific components: a parameterized set of behaviors which can emulate the behavior of the system element represented by the actor, and an internal representation of the external system which can be used to estimate the fitness of trial sets of the autonomous cognitive element's control parameters. The generation and adaptation of the trial control parameter sets is accomplished in application-independent elements of the cognitive architecture.

A specific simulation actor has been developed to represent the ground station module that evaluates JSTARS images of vehicles on battlefields. The cluster location, characterization and discrimination processes performed by the intelligent human GSM operator were implemented into a parameterized set of behaviors by using a newly developed family of 2D cluster filters. This GSM actor has demonstrated the ability not only to recognize military formations under prescribed conditions, but to adapt its behavior to unanticipated conditions that develop in the complex simulated battlefield system.

9. REFERENCES

1. H. Simon, The Sciences of the Artificial, MIT Press,Cambridge, Mass. (1981)
2. M. M. Waldrop, Complexity, Simon and Schuster, N.Y., (1992).
3. Field Manual No. 34-25-1: Joint Surveillance Target Attack Radar System (Joint STARS), unlimited distribution from Department of the Army Headquarters, Washington D.C. (1995)
4. for example, C. R. Karr, D. Reece, R. Franceschini, "Synthetic Soldiers," IEEE Spectrum, vol. 34, no. 3 (1997).
5. A. Newell, Unified Theories of Cognition, Harvard University Press, Cambridge, Mass, (1990).
6. R. P. Minch, "Toward a Parsimonious Architecture for Intelligent Organizational Information Systems," proc. 25th Hawaiian Int'l Conf. System Sciences, Vol. IV, pp 454-463, (1992).
7. H. Plotkin, Darwin Machines and the Nature of Knowledge, Harvard University Press, Cambridge, Mass. (1994).
8. P. D. Stroud, "Simulation-based Learning in Knowledge-based Controllers," Proc. 1996 IEEE Int'l Symposium on Intelligent Control, pp. 168-174, (Sept 1996).
9. J. H. Holland, Adaptation in Natural and Artificial Systems, 2nd ed., MIT Press,Cambridge, Mass. (1992).
10. D. Gabor, "Theory of Communication J IEE, Vol. 93, pp. 429-459 (1946).
11. X. T. Oliveira e Silva , H. J. W. Belt, "On the determination of the optimal center and scale factor for truncated Hermite series," Proc. EUSIPCO-96, eds. G. Ramponi, G.L. Sicuranza, S. Carrato, S. Marsi; Edizioni LINT Trieste srl, Vol. III, pp. 1563-1566. (1996).
12. J. Yang, J. Opt. Soc. Am, Vol. 9, No. 2, pp. 333-336 (1992)
13. U. Pen, "Discretized Hermite polynomial expansion on Modes of elliptical galaxies," Astrophysical J., Vol. 43, p. 104 (1994).
14. Y-H. Gu, "Adaptive multiresolution Hermite-binomial filters for image edge and texture analysis," proc. SPIE Vol. 2308, p.748, (1994).
15. M. Basu, and L. M. Kennedy, "An experiment with Gaussian derivatives for image enhancement," proc. IEEE Int. conf. on Systems, Man and Cybernetics, Vol. 4, p.3778-3783 (1995) -models eye with Hermite filters
16. J. Martens, "The Hermite transform - theory," IEEE trans. on Acoustics, Speech, and Signal Processing, Vol. 38, No. 9, pp. 1595-1606 (1990).
17. I. Gertner and G. A. Geri, "Image representation using Hermite functions," Biol. Cybern., Vol. 71, pp. 147-151 (1994).
18. Y. V. Venkatesh, K. Ramani, R. Nandini, "Hermite Sieve as a wavelet-like array for 1D and 2D signal decomposition," IEE proc. Vision, Image, and Signal Processing, Vol. 141, p. 348 (1994)
19. M. Born and E. Wolf, Principles of Optics, sixth ed., Pergamon Press, Oxford (1980)
20. Handbook of Mathematical Functions, M. Abramowitz and I. A. Stegun, eds., Dover.
21. R. J. Noll, "Zernike polynomials and atmospheric turbulence," J. Opt. Soc. Am, Vol. 66, No. 3, pp. 207-211 (1976).
22. V. V. Dodonov, and V. I. Man'ko, "New Relations for two-dimensional Hermite polynomials," J. Math. Phys., Vol. 35, No. 8, pp. 4277-4294 (1994).